CONFIDENTIAL

CHIVE/C-102-66
4 May 1966

MEMORANDUM FOR: Director of Computer Services

SUBJECT: System /360 Problems

1. The purpose of this memorandum is to call your attention to some problems related to the IBM System/360, to indicate the hazards that we may face in the future with this system, and to recommend remedial and preventive action.

2. The following comments are based on efforts to train both experienced and new programmers and to use the Model 30 for a rather straightforward application--production of a permuted title index.* Five programmers have been involved with this activity, all with some programming experience on other machines. (Fewer people were actually needed, but we used the opportunity to expose several people to the 360.) All of the basic programming for this job was done in January and February. The program is still not checked out. In a stable situation with familiar equipment and software, it would have been completed in one month. The specific reasons for the delay could be enumerated in detail here, but this is less important than recognizing in this prototype situation a group of problems which, if not addressed now, could have adverse effects on the performance of OCS in the future.

3. The basic problem, of course, is that the 360 is new, both to IBM and their customers. OCS has never had the experience of being "first" (even the ALP System has a considerable history). Success in being "first" requires a management perspective somewhat different than that needed in a stable operational situation: it should be recognized that in committing itself to the 360, all of OCS in a sense now becomes a development activity.

* For the most part, these comments have been confirmed by Applications Division's initial 360 experience (redesign and conversion of the Agency Training Records application).

CONFIDENTIAL

A significant portion of OCS resources and management time must be devoted to internal activities related to the 360 that will have no direct payoff to customers for some time.

4. Newness in itself perhaps can be handled, but a more serious concern is that the shakedown period for the 360 may be longer than that of any computer previously put on the market. It took several years to bring the IBM 704, 709, 7090 series into really effective use; the SHARE activity on these machines involved thousands of programming man-years. OCS had the advantage of results of all of this effort from the start. If we can assume that the 360 series (and its associated software) is at least as complex and powerful as the 7000 series, we should expect the 360 shakedown period to last at least four years.

5. More specific aspects of the problem--as CHIVE has already encountered them--are mentioned below from the point of view of the applications programmer, software, and computer operations.

a. Applications Programmer. In general, programmers with experience in either the 7090 or the 1410 find 360 programming more difficult to master. Several characteristics of the 360 "architecture" cause problems: the base-plus-displacement addressing and the 4096 byte/block boundaries are difficult to cope with; several important instructions require careful alignment of data on word boundaries; the instruction reportoire is large and complex; the interrupt structure is powerful, but at least as difficult as the 7090. Again, newness may be the problem; there are several new concepts (or new ways of looking at familiar concepts) that the programmer must master.

It might be argued that higher-level programming languages (when they become available) will alleviate the need for detailed programmer knowledge of machine characteristics. This may be true in certain applications, but it should be pointed out that, while such programming languages are becoming more powerful, the applications programmer is becoming more sophisticated at the same time and will always have need for access to basic machine facilities.

For the inexperienced programmer, learning the 360 is perhaps more of a nightmare than he would face learning programming on another machine. This problem is aggravated by the inadequacy of the training facilities available for the neophyte, and becomes almost impossible when the burden of additional training required for proper software and language utilization is added.

Perhaps the most serious problem faced by the applications programmer is that he is expected to fully exploit the additional power available in the 360. These facilities have been well advertised, and our customers can properly demand better performance over a wider range of applications (with the use of terminals, direct access, etc.).

b. Software. This new software philosophy requires that it operate hand-in-hand with basic machine capabilities. This adds a third party to the team with the attendant communication problems. Now the programmer can blame not only the machine, but software for his program failures and vice versa. This is a serious problem right now; we spend days trying to determine causes of failure.

Secondly, this ambitious philosophy causes problems in even the most fundamental operations during this software buildup period. The framework for the operating system is being designed to handle every conceivable situation, because of the complex framework, utility packages in the software that otherwise would be straightforward and efficient are now ponderous and inefficient. Evidence of this factor includes extremely large storage overhead for bookkeeping in storage devices and indications that assembly time on the Model 65 will be longer than on the 7090 for an equivalent program. The cataloging feature used in conjunction with the assembly program is of little use during the buildup period and is actually causing more problems than it is solving. A basic debugging aid, a "character" memory dump, is not yet available to the programmer.

The basic point here is that the software appears to be over designed for our more immediate needs. The ultimate implementation of the operating system may prove

the need for several of the planned "bells and whistles," but this does not solve our problem in the interim period.

We continue to hear about internal IBM problems in software implementation. For example, there are rumors now of additional slippages, and the senior CHIVE IBM programmers returning from the best course on the operating system at Poughkeepsie indicated that no one person there really understands the complete operating system.

The Technical Staff is now heavily burdened with a variety of problems in coping with 360 software--understanding the basic philosophy; coping with inadequate documentation; incorporating changes received from IBM; determining whether a failure is due to hardware, software, or an applications program; and communicating this knowledge to the rest of the Office. In addition, it recently assumed the task of building an in-house time-sharing capability. Applications programmers are sympathetic to this work burden and are reluctant to seek assistance from the Technical Staff which is now needed on a daily basis.

c. Operations. Again, unfamilarity with a new system--both in terms of software and hardware--is at the root of some operational problems which we have encountered. There is no substitute for long-term, hands-on experience in learning how to get the most out of the computer at the console and in diagnosing specific failures on the spot to assist programmers and engineers. The instability of the software causes additional operational problems. The 360 console is at least as complex as a 7090.

Several problems we have encountered in operations might be noted here: the operators' lack of knowledge of the assembler sometimes results in abortive runs due to error messages that could be bypassed, or due to failure to reinitialize the assembler; in many instances, incomplete documentation was returned to the programmer after a test run because the operator did not know what went wrong. Severe delays could have been avoided if the operator knew how to correct a seemingly abortive situation on the spot and rerun the job.

The problems in dealing with a new machine could be better handled if there were enough operators so that some could be assigned specifically to the 360 to get experience in unusual operating problems and pass on their knowledge to others. There appears to be little continuity now in applying such experience. Finally, it appears to be inappropriate to operate the 360 during this developmental period on a closed-shop basis. Efficient communications during this period are of utmost importance and could be encouraged by intimate use of a three-man team in all 360 operations--applications programmer, systems programmer, operator.

6. Recommendations. Short of not being "first"; i.e., delaying 360 implementation or substituting other more reliable and familiar equipment, a number of actions could be taken to insure that we are using our resources in the most profitable manner in getting the 360 to work efficiently for us:

a. The possibility of acquiring one or more 360 systems dedicated to assembly, compilation and debugging should be considered. It might be argued that in theory this is not necessary because of the inherent speed and power of the Model 65 to be delivered. This argument can be answered with three points: First, the need for open-shop operating during the period when the reliability of the software can be questioned; second, on big machines, small errors in stacked test runs can cause big, costly aborts; and third, the availability of a multi-task capability will not have a material effect on test run setup time (it will take just as long to move cards from one place to another and to mount tapes).

The acquisition of one or more such systems, of course, cannot be justified on the basis of production requirements, but I would estimate that about 100 programmers will soon be requiring almost daily access to the 360 for testing training programs, conversion problems, and new applications. The programmer man-hours saved in avoiding long turn-around delays could justify the additional rental cost.

b. Some changes in the policy of the Operations Division should be considered (implementation of some of these would require additional personnel): interim open-shop operations on the 360, rather exhaustive training of operators in 360 programming and operating systems, the assignment of a set of operators to specialize on the 360, good policing of the 360 schedule to permit the three-member team to be present during program test, and consideration of turn-around time, as well as throughput, as criteria for efficient computer operations.

c. Additional resources should be provided for the Technical Staff to permit more dynamic communications with operators and applications programmers. Such resources might take the form of personnel specifically assigned to software documentation and quick delivery of notification of changes to the above people, full-time Technical Staff members at IBM software programming sites; more definitive communications with IBM in terms of memoranda demanding specific answers to specific questions; meeting software requirements as voiced by applications programmers.

d. OCS conversion plans and schedules should be re-examined with consideration given to improving pre-360 operations on the 7010, 7090, 501, and 301 and to delaying conversion of major OCS programs.

e. Adequate 360 training should be given top priority now and the necessary Agency resources allocated to plan and execute such a training program.

f. Candid discussions with customers of OCS services should be held so that they can determine the effects of possible delays in 360 capability. Such discussions should err on the side of pessimism to avoid OCS commitment to a potentially untenable program.

[redacted]
Director
CHIVE Task Force

25X1A

cc: Chief, Appl. Div.
Chief, Opr. Div.
Chief, Tech. Staff